Workshop Proposal for EuroSys 2014

Principles and Practice of Eventual Consistency (PaPEC)

Marc Shapiro, INRIA & LIP6
Nuno Preguiça, UNL
Alexey Gotsman, IMDEA Software Institute
Peter Bailis, UC Berkeley
Justin Sheehy, Basho Inc.

18 October 2013

1 Abstract

Strong consistency is easy to understand, but the synchronisation it requires has a high cost and does not tolerate partial system failure or network partitions well. Hence, the past decade has seen renewed interest in an alternative —Eventual Consistency (EC)— where data is accessed and modified without coordination. Many production systems guarantee only EC, including NoSQL databases, key-value stores and cloud storage systems. EC improves responsiveness and availability, updates propagate more quickly, and enables less coordination-intensive and often less complex protocols.

EC is crucial for scalability but represents an unfamiliar design space. Conflicts and divergence are possible, making them hard to program against, while metadata growth and maintaining system-wide invariants can also become problematic. As more developers interact with and program against distributed storage systems, understanding and managing these trade-offs becomes increasingly important.

Although EC remains poorly understood, there has recently been considerable momentum in the research and development community: the past several years have seen the (re)introduction of useful concepts, such as replicated data types, monotonic programming, causal consistency, red-blue consistency, novel proof systems, etc., designed to allow improve efficiency, programmability, and overall operation of weakly consistent stores.

This workshop aims to investigate the principles and practice of Eventual Consistency. It will bring together theoreticians and practitioners from different horizons: system development, distributed algorithms, concurrency, fault tolerance, databases, language and verification, including both academia and industry. In order to make EC computing easier and more accessible, and to address the limitations of EC and explore the design space spectrum between

EC and strong consistency, we will share experience with practical systems and from theoretical study, to investigate algorithms, design patterns, correctness conditions, complexity results, and programming methodologies and tools.

2 Summary, scope and significance

Interest in eventual consistency has surged recently from the development of global-scale distributed systems and of applications for mobile devices; witness papers at major venues such as SOSP, POPL, NSDI, CIDR, etc.

The well-known CAP theorem shows that in a large-scale distributed system, where networks routinely fail, a system must choose between "strong" consistency (such as linearizability or serializability) or availability ("always on" operation). As high availability is a requirement for many Internet services, a new generation of distributed systems has embraced weaker consistency models. Even in the absence of failures, weaker consistency improves latency, since it removes a costly synchronisation bottleneck. As updates do not need to coordinate, applications can execute in parallel, with low latency, on replicas close to their users (enabled by geo-replication).

The inherent trade-off between consistency, availability and the ability to tolerate failures remains poorly understood but has significant implications for system designers and application developers. As distributed computing becomes mainstream, we need to improve our understanding, and to develop abstractions and tools, in order to enable programmers who are not experts to use them.

In particular, EC is hard to apprehend by application programmers, since data may diverge, conflict, and roll back. A common approach is *Last-Writer-Wins (LWW)*, which deterministically installs the update with the highest timestamp at all replicas, and simply discards the other updates, which is problematic in a number of cases. Alternatively, the system (e.g., Dynamo or Riak) can maintain multiple concurrent versions, and leave responsibility to the application to choose or merge an unified version, possibly by asking the user. A different approach is to specify algorithmically the composition of concurrent updates, as in operational transformation (OT) or conflict-free replicated data types (CRDTs).

A Dagstuhl seminar recently took place to make progress towards this improved understanding [1]. It brought together approximately 50 researchers and practitioners in the areas of distributed systems, programming languages, databases and concurrent programming. Its aim was to understand lessons learnt in building scalable and correct distributed systems, the design patterns that have emerged, and explore opportunities for distilling these into programming methodologies, programming tools, and languages to help make distributed computing easier and more accessible. The current proposal is in some sense a continuation of the Dagstuhl workshop, focusing more specifically on EC.

Workshop topics include:

- Design principles, correctness conditions, and programming patterns for EC.

- Techniques for eventual consistency: session guarantees, causal consistency, operational transformation, conflict-free replicated data types, monotonic programming, state merge, commutativity, etc.
- Consistency vs. performance and scalability trade-offs: guiding developers, controlling the system.
- Analysis and verification of EC programs.
- Strengthening guarantees: transactions, fault tolerance, security, ensuring invariants, bounding metadata memory, and controlling divergence, without losing the benefits of EC.
- Platform guarantees vs. application involvement: guiding developers, controlling the system.

This workshop is relevant to EuroSys, since it is about approaches for building large-scale, efficient and fault tolerant systems.

3 Procedures and format

The workshop will feature a small number of invited talks. Participants will submit a short (5-page) position paper in response to an open CFP. The PC will review the position papers, and select the most interesting ones for presentation.

Each workshop session will contain a number of presentations around a common theme, followed by a panel and a general discussion on the theme.

The papers and slides will be distributed to the participants.

Based on the Dagstuhl participation, we expect 30 to 40 participants.

4 Brief bios

4.1 Primary contact: Marc Shapiro

Marc Shapiro, Senior Researcher
INRIA & LIP6
Bureau 26-00/211, Boîte courrier 169
4, place Jussieu
75252 Paris Cedex 05
France
Tel.: +33 1 4427 7093
e-mail: Marc.Shapiro@acm.org
http://lip6.fr/Marc.Shapiro/

Marc Shapiro is a Senior Researcher (Directeur de Recherche) at Inria. He has been a Principal Investigator for many research projects, both academic and in collaboration with industry. He spent six years and a half at Microsoft Research Cambridge as Senior Researcher, managing the ten-person Cambridge Distributed Systems Group (Camdis).

Marc Shapiro has a long history of research and publication in distributed systems, including top-tier conferences such as SOSP, OSDI, PODC, PLDI, ICDCS, PPoPP, SRDS, and SSS. Marc Shapiro is the Principal Investigator of the ConcoRDanT project (ANR, France), and of the SyncFree European project (FP7), both on developing technology for eventual consistency.

4.2 Nuno Preguiça

Nuno Preguiça, Associate Professor
Departamento de Informática
FCT, Universidade Nova de Lisboa
Quinta da Torre
2829-516 Caparica
Portugal
nuno.preguica@di.fct.unl.pt
http://www-asc.di.fct.unl.pt/~nmp/

Prof. Nuno Preguiça has received a PhD degree from UNL (2003). Since 2003 he is Assistant Professor at UNL and a Research at CITI. In 2011 he spent his sabbatical leave at Inria. He has received a Google Research Award in 2009 by his work on solutions for cloud data management. His primary research interests have been on data management, currently focused on cloud computing. He has been the principal investigator of national projects FEW, Byzantium, RepComp and participated in several other projects.

4.3 Alexey Gotsman

Alexey Gotsman
Assistant Research Professor
IMDEA Software Institute
Campus Montegancedo s/n
28223-Pozuelo de Alarcón
Madrid
Spain
Tel.: +34-91-101-2202 ext 4136
Fax: +34-91-101-1358
Alexey.Gotsman@imdea.org

Alexey is a tenure-track Assistant Research Professor at the IMDEA Software Institute. Before joining IMDEA, he was a postdoctoral fellow at the University of Cambridge, where he also got his Ph.D. His thesis received a Best Dissertation Award of the European Association for Programming Languages and Systems (EAPLS). Alexey's research interests are in software verfication, particularly, in developing reasoning techniques and automated verification tools for real-world concurrent systems software. He is currently coordinating a EU FET Young Explorers project ADVENT.

4.4 Peter Bailis

Peter Bailis, PhD student
UCB EECS
387 Soda Hall
MC 1776
Berkeley, CA 94704
USA
http://www.bailis.org/
pbailis@cs.berkeley.edu

Peter Bailis is a graduate student of Computer Science in the AMPLab and BOOM projects at UC Berkeley, where he works closely with Ali Ghodsi, Joe Hellerstein, and Ion Stoica. He currently studies distributed systems and databases, with a particular focus on distributed consistency models and their programmability. Peter received his A.B. from Harvard College and is the recipient of the NSF Graduate Research Fellowship and the Berkeley Fellowship for Graduate Study.

4.5 Justin Sheehy

Justin Sheehy, Chief Technology Officer
Basho, Inc.
700 Massachusetts Ave, 2nd Floor
Cambridge, MA 02139
USA
Phone: +1 (617) 714-1700
www.basho.com
justin@basho.com

Justin is Chief Technology Officer at Basho, where he directs technical strategy and new research into storage and distributed systems.

Prior to Basho, Justin was a Principal Scientist at MITRE where he led research projects in topics including high assurance platforms, automated defensive response, cryptographic protocol analysis, mission assurance, and resilient systems. Before MITRE, Justin worked at a series of technology companies including five years at Akamai Technologies, where he was a key contributor to the technology that enabled fast growth of Akamai's infrastructure.

Justin's undergraduate and graduate studies in Computer Science were performed at Northeastern University.

5 Important dates

Tentative dates:

- Submission deadline: 27 January 2014
- Notification date: 28 February 2014

References

[1] Bettina Kemme, Ganesan Ramalingam, André Schiper, Marc Shapiro, and Kapil Vaswani. Consistency in distributed systems (Dagstuhl seminar 13081). *Dagstuhl Reports*, 3(2):92–126, 2013. ISSN 2192-5283. doi: http://dx.doi.org/10.4230/DagRep.3.2.92. URL http://drops.dagstuhl.de/opus/volltexte/2013/4014.